REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE (DD-MM-YYY) 31-October-2009	2. REPORT TYPE Performance/Technical Report	3. DATES COVERED *(From – To)* 22-Feb-2009 to 30-Jun-2009

4. TITLE AND SUBTITLE

INTEGRATED RECONFIGURABLE INTELLIGENT SYSTEMS (IRIS) FOR COMPLEX NAVAL SYSTEMS

5a. CONTRACT NUMBER

N00014-08-1-0107

5b. GRANT NUMBER

N/A

5c. PROGRAM ELEMENT NUMBER

N/A

6. Author(s)

Dr. Dimitri N. Mavris

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Georgia Institute of Technology
School of Aerospace Engineering
Atlanta, GA 30332-0150

8. PERFORMING ORGANIZATION REPORT NUMBER

N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Office of Naval Research
875 North Randolph Street
Arlington, VA 22203-1995

10. SPONSOR/MONITOR'S ACRONYM(S)

ONR

11. SPONSORING/MONITORING AGENCY REPORT NUMBER

N/A

12. DISTRIBUTION AVAILABILITY STATEMENT

Unlimited Distribution

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The report details the research work has been done by ASDL in developing and applying the IRIS concept for the period of February 22 to June 30, 2009. The main objective of the work for this period is to further develop and refine the integrated modeling and simulation environment in order to investigate the behavior of complex naval systems for improving the ship design and operations. Five individual tasks were conducted to fulfill this objective. Models revised based on the notional YP were integrated and tested; a control architecture with inference engine was proposed and evaluated based on defined scenarios; comparison of plain NN models and NN models with the block-structure was performed for evaluating accuracy of the surrogate models; a suitable framework and a database engine has been selected to facilitate the information management for developing the HMI of the simulation environment; a robust and a resilient approaches were proposed to conduct design space exploration in order to obtain an ultimate design with increased survivability and mission effectiveness. The rest of the report will explain the accomplishments for each task in details.

15. SUBJECT TERMS

Modeling & Simulation, Reconfigurability, Integrated & Intelligent, Naval Systems

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Mr. ANTHONY J. SEMAN
					19b. TELEPHONE NUMBER *(include area code)*
U	U	U	SAR	16	703-696-5992

TECHNICAL REPORT

(February 22, 2009 – June 30, 2009)

"Integrated Reconfigurable Intelligent Systems (IRIS) for Complex Naval Systems"

Contract #: N00014-08-1-0107

SUBMITTED TO:
Mr. Anthony J. Seman, Office of Naval Research
(email: Anthony_Seman@onr.navy.mil)

Office of Naval Research
875 North Randolph Street
Arlington, VA 22203-1995

SUBMITTED BY:
Georgia Institute of Technology
School of Aerospace Engineering
Aerospace Systems Design Laboratory (ASDL)
Atlanta, Georgia 30332-0150

AWARD PERIOD:
December 31, 2007 to June 30, 2009

October 31, 2009

Principal Investigator:
Professor Dimitri N. Mavris
Director
Aerospace Systems Design Laboratory
School of Aerospace Engineering
Georgia Institute of Technology
Phone: (404) 894-1557
dimitri.mavris@ae.gatech.edu

Summary

The following report details the research work has been done by ASDL in developing and applying the IRIS concept for the period of February 22 to June 30, 2009. The main objective of the work for this period is to further develop and refine the integrated modeling and simulation environment in order to investigate the behavior of complex naval systems for improving the ship design and operations. Five individual tasks were conducted to fulfill this objective. Models revised based on the notional YP were integrated and tested; a control architecture with inference engine was proposed and evaluated based on defined scenarios; comparison of plain NN models and NN models with the block-structure was performed for evaluating accuracy of the surrogate models; a suitable framework and a database engine has been selected to facilitate the information management for developing the HMI of the simulation environment; a robust and a resilient approaches were proposed to conduct design space exploration in order to obtain an ultimate design with increased survivability and mission effectiveness. The rest of the report will explain the accomplishments for each task in details.

Task 1: Integration of Heterogeneous Systems

In order to represent a notional YP ship in a computer simulation environment, the models of sub-components must be created and integrated into a single model that represents the ship. The sub models used were created by several members of the GT IRIS team. They comprise of a power model for the representation of the electric system, a fluid model that represents the cooling fluid flow, three layers of controllers, a scenario script, and a Human-Machine Interface.

The power model is modeled in Simulink. It is set up modularly, and the number of loads is easily changeable. It is a physics-based model, and has the main physical components that an actual power model would include, such as controllers, busses, service loads, etc. The power model creates the thermal loads to be taken care of by the fluid flow model. The cooling fluid flow is modeled in Flowmaster V7. It represents the cooling flow network, and interacts with the service load temperatures from the power model. The fluid model also has a damage mode modeled in. This is simply represented by two valves that open into the environment, thus simulating a pipe rupture, similar to what would be the outcome of a missile hit of the ship.

The opening of the rupture valves is controlled by the scenario script, which essentially determines when a rupture happens, and where. Currently, ruptures are only implemented at one location in the fluid model, but more will be modeled in a next effort.

The three controllers are responsible to correctly react to ruptures within the fluid flow model and distribute the cooling fluid to the service loads. The controllers detect the rupture location, and shut down valves accordingly such that the rupture within the flow network is isolated, and cooling fluid loss is prevented.

The sub models were integrated using Phoenix Integration ModelCenter. A scheduler script was written, which takes care of running the models and exchanging the necessary data between the respective models over the time. The script also allows for temperature unit conversions between some models. It further allows for arbitrarily scheduled execution of sub models, a feature that proved to be necessary for correct execution and communication between the low-level and mid-level controllers. Also, models can be exchanged and modified more easily when integrated with a script instead of the ModelCenter link editor. Lastly, the scheduler allows for adaptive time steps, a feature that will be implemented once the simulation runs correctly.

A makeshift interface for data output and visualization was included. The output is into an Excel worksheet. It allows for free choice and selection of variables to be displayed in standard Excel graphs. However, data is only stored until the next simulation run. More importantly, the interface is read-only, which means that the simulation cannot be influenced during the run, the Excel interface only helps to display variables. A much more sophisticated database driven Human Machine Interface (HMI) is currently under development. A preliminary version has been shown to work successfully with the simulation.

Previous investigations into general methods of optimizing simulation execution time and accuracy have shown that an adaptive simulation time step is beneficiary to both real world execution time and result accuracy. Related publications by Nairouz and Hoepfer have been cited previously. The approaches used for these publication was a crude rule of thumb algorithm, with no further evaluation regarding optimized parameters etc. Hence, further efforts were made to investigate this issue. This leads to the more general question of model evaluation and optimization. It must be kept in mind that in a real simulation, such as the one that will eventually model the actual YP ship, consists of dynamic sub models whose properties are unknown. Hence, it must be determined whether the outputs from the integrated co-simulation actually represent the real world system output. Time step optimization will then be a sub problem to this general problem. First steps have been made to evaluate which approaches seem appropriate to determine the actual system output. Ideally, the applied algorithm would determine, from the current simulation step, the next simulation step and the system states at that point. It also would determine an error bound within which it is assumed that the function accurately represents the real world system. If the system stays within this error bound, it is deemed to be accurate, and a next time step can be evaluated. If the system goes beyond the error bound, then it is assumed that it does not accurately represent the real world system. A deviation from the error bounds might also indicate an external shock input to the system. In either case, corrective actions must be determined which will need to be taken in order to get the simulation back into accurate real world representation. In the case of error deviation due to internal model behavior, a reduced time step may be an accurate and simple solution. In the case of external shock, such as sudden system alteration due to ruptures etc., the case is different, since the simulation must be able to handle such instances, and re-configure the system accordingly. Hence, the corrective action necessary will need to be more elaborate.

For simple simulation of an integrated model, mathematical models have been investigated that might help to determine the correctness of the simulation outputs. These methods are based on numerieal methods for the solution of differential equations. If a differential equation is to be solved on a digital eomputer, it may not be able to solve the equation and use the solution to determine the "path" of the equation variables. Henee, numerieal methods are used to solve sueh equations. The main property that links these methods to the simulation of an integrated model is that, for both eases the aetual equations are unknown and henee need to be approximated. First approaehes and algorithms have been identified, and are eurrently being implemented on a simple test model. Further literature review showed that there are more sophistieated methods for evaluation, whieh represent predietor-eorreetor methods. These methods first estimate the next time step point, and then use eorreetive measures to determine the aeeuraey of the point. Also, some methods have adaptive time steps ineluded, whieh keep the simulation within desired error bounds. However, these methods require large eomputational expenses, and an investigation may be necessary to determine the tradeoff between aeeuraey and eomputation time.

Task 2: Multi-Agent Based Mid-level Control with Dynamic Inference Engine

Introduction

Inereasing soeietal demand for automation has led to eonsiderable efforts for eontrolling large-scale complex systems, especially in the area of autonomous intelligent control methods. A eontrol system of a large-seale eomplex system needs to satisfy four system level requirements: robustness, flexibility, reusability, and sealability. Corresponding to the four system level requirements, there arise four major ehallenges of eontrolling large-seale eomplex systems. First, it is diffieult to get aeeurate and eomplete information. Seeond, the system may be physieally highly distributed. Third, the system evolves very quiekly. Fourth, emergent global behaviors of the system ean be eaused by small disturbanees at eomponent level. To deal with those ehallenges, Hybrid Multi-Agent Based Control (HyMABC) arehiteeture with Multiple Seetioned Dynamie Bayesian Networks (MSDBNs) inferenee engine have been proposed.

Hybrid Multi-Agent Based Control (HyMABC) Architecture

HyMABC arehiteeture eombines hierarehieal eontrol arehiteeture and module eontrol arehiteeture together to form a hybrid eontrol arehiteeture. First, it deeomposes a eomplex system hierarehically; second, it eombines the eomponents at the same level as a module and then designs eommon interfaees for all of the eomponents in the same module; third, a few replieations are made for eritieal agents and are organized into some logical rings. It keeps elear guidelines for eomplexity deeomposition and also reduees eommunieation eomplexity of the distributed eontrol system.

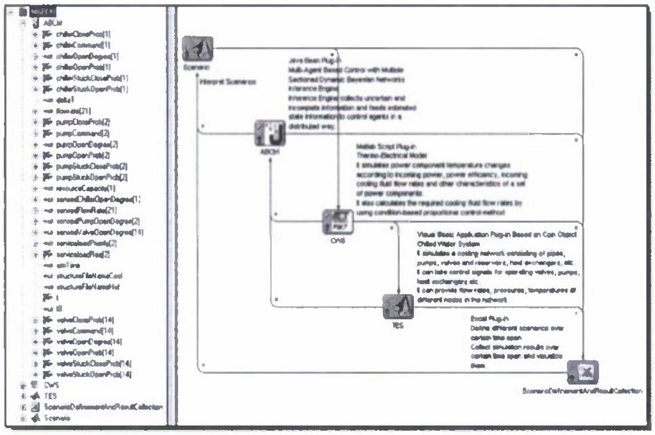
For an important control agent such as the highest level agent in the multi-agent system, if it is damaged or unavailable, the whole system will lose the global control even though the subsystems ean work aeeording to their available information. In order to keep the whole system working and prevent a failure of signifieant eontrol agent from oeeuring, a few replieations are ereated and arranged in a robust and effieient way to insure automatic reconfiguration when necessary. Similar to the idea of fault-tolerance with replicated main containers in Java Agent DEveleopment (JADE), it starts with many replications of a significant control agent as needed. The replicated agents are software based and modulated, thus it is easy to apply. All of the replications arrange themselves into a logical ring. Whenever one of the replications fails, the others will notice and act accordingly by using cross-notification. Agents connecting to the failed replication will be able to connect to some other replications and keep all of the information as the same as before the damage happens.

Multiple Sectioned Dynamic Bayesian Networks (MSDBNs) Inference Engine

Multiple Sectioned Dynamic Bayesian Networks (MSDBNs), as a distributed, dynamic, probabilistic inference engine, can be embedded into the control architecture to handle uncertainties of general large-scale complex systems. MSDBNs decomposes a large knowledge-based system into many agents. Each agent holds its partial perspective of a large problem domain by representing its knowledge as a dynamic Bayesian network. Each agent accesses local evidence from its corresponding local sensors and communicates with other agents through finite message passing. If the distributed agents can be organized into a tree structure, which satisfies running intersection property and d-sep set requirements, globally consistent inferences are achievable in a distributed way. By using different frequencies for local DBN agent belief updating and global system belief updating, it balances the communication cost and inference global consistency. In this research, fully factorized Boyen-Koller (BK) approximation algorithm is used for local DBN agent belief updating, and static Junction Forest Linkage Tree (JFLT) algorithm is used for global system belief updating.

Modeling and Simulation Environment

Multi-agent based control model with distributed multiple sectioned dynamic Bayesian network inference engine has been established for a simplified chilled water system. This simplified chilled water system includes one chiller-pump plant and two service loads.



Java Bean Plug-in
Multi-Agent Based Control with Multiple Sectioned Dynamic Bayesian Networks Inference Engine
Inference Engine collects uncertain and incomplete information and feeds estimated state information to control agents in a distributed way.

Matlab Script Plug-in
Thermo-Electrical Model
It simulates power component temperature changes according to incoming power, power efficiency, incoming cooling fluid flow rates and other characteristics of a set of power components.
It also calculates the required cooling fluid flow rates by using condition-based proportional control method.

Visual Basic Application Plug-in Based on Con Object
Chilled Water System
It simulates a cooling network consisting of pipes, pumps, valves and reservoirs, heat exchangers, etc.
It can take control signals for operating valves, pumps, heat exchangers etc.
It can provide flow rates, pressures, temperatures of different nodes in the network.

Excel Plug-in
Define different scenarios over certain time span
Collect simulation results over certain time span and visualize them

Figure 1: The Entire Test Model in ModelCenter Analysis View

An integration environment shown in Figure 1 has been developed by using ModelCenter® of Phoenix Integration to test the proposed methods. The integration model includes five modules: Scenario, ABCtrl, CWS, TES and ScenarioDefinementAndResultCollection. Scenario module transforms the scenarios defined in ScenarioDefinementAndResultCollection module into the format which is compatible with CWS module created in Flowmaster. ABCtrl includes HyMABC and MSDBN, which consist of dozens of control agents and three Bayesian network agents. All of the agents are established in JADE which is completely implemented in Java language, while CWS simulates fluid network which balances energy, pressure and mass flow rate of fluid. TES is a thermoelectric model developed in MATLAB Simulink and it also includes low level feedback controllers. ScenarioDefinementAndResultCollection is implemented in Excel worksheet. It defines the scenarios, collects the simulated results and visualizes the results.

Result Analysis

By using the integrated model, three scenarios have been tested and analyzed.

- **Scenario 1 (Nominal Conditions)**:

Assumptions: all of the components are not damaged; every flow rate point in the Bayesian network is observable; every component open degree is observable; resource capacity is 0.8kg/sec; the initial temperatures of service load 1 and service load 2 are 317 Kelvin and 400 Kelvin respectively. For the nominal case, the control system can make the right decisions and distribute the resource to different service loads accordingly.

- **Scenario 2:**

Assumptions: all of the flow rates listed in the Bayesian network are not observable; every component open degree is observable; resource capacity is 0.8kg/sec; valve7 becomes STUCKCLOSE at time $t = 440\sec$ (the 11th iteration); valve11 becomes STUCKCLOSE at time $t = 840\sec$ (the 21st iteration); resource capacity is 0.8kg/sec; the initial temperatures of service load 1 and service load 2 are 317 Kelvin and 400 Kelvin respectively. For this case, the results shows that without any flow rate observation and only with component open degree observations, the inference engine can detect component damages quick enough and the control system can reconfigure the whole system by switching from damaged components to their corresponding redundant ones to redistribute system resource accordingly.

- **Scenario 3:**

Assumptions: only the flow rates of the points located in the upstream of valves in service loads and listed in the Bayesian networks are observable; valve open degrees are observable only for valve1, valve2 and valve7. Pumps and chiller operation states are observable. Valve 7 is STUCKCLOSE at time $t = 440\sec$ (the 11th iteration). Valve 11 is STUCKCLOSE at time $t = 840\sec$ (the 21st iteration); resource capacity is 0.8kg/sec; the initial temperatures of service load 1 and service load 2 are 450 Kelvin and 400 Kelvin respectively. For this case, the results show that it is hard to detect component state changes by only using flow rate observations, which is due to cyclic characteristic of the fluid network. Fortunately, for a smart valvc, its open degree is one of the output signals.

Task 3: Surrogate Modeling of Dynamic Systems

After the development of the surrogate modcl with a block structured NN and its successful implementation to simple nonlinear RLC modeling as proof-of-concept, thc dynamic surrogate modeling method based on the bloek structured NN is eurrcntly being used for creating component models of the chilled-water model of the notional YP. In this report, one of the component models was ehosen in order to eonduet a performanee eomparison between the surrogate model with plain single hidden layer (SHL) NN and the surrogate model with the block-structured NN. The model is shown in Figure 2 and its speeifieation is listed in Table 1.

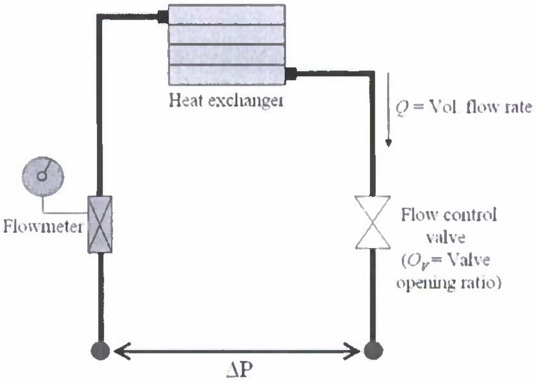


Figure 2: Model of a Heat Exchanger Unit

Table 1: Model Specification

Pipeline	*Heat exchanger*	*Valve*
Tot. pipe length: 12 ft Diameter: 0.5 in	Pipe area: 0.197 in2 Loss coefficient: 1.5 Hydraulic diamcter: 0.5 in	Diameter: 0.5 in

In Figure 2, Q is volumetrie flow rate, O_v denotes valve opening ratio, and ΔP represents the pressure differenee between the two ends of the system. The model was originally ereated using FlowMaster® V7.

Table 2 shows the differences in the structural configuration of the two NNs. The major difference is that the bloek-struetured NN has two hidden layers instead of one, and the neurons at the first layer have the linear aetivation function. The number of nodes at the linear layer (i.e., the first layer) should be the same as the dimension of the system state variables (and the dimension of the final outputs from thc NN in this formulation), so only one node was assigned. A graphieal representation of the bloek-struetured NN will be, as a result, very similar to that shown in Figure 3, in whieh a "bottleneck" structure ereated by the linear layer is elearly found. The same number of nodes was assigned in the nonlinear layers of both NN structures so that the two NNs have at least the same potential capability of nonlinear function approximation.

Table 2: NN Structure

	Plain NN	Block structured NN	
Net structure:	Single hidden layer	Double hidden layer	
		Layer 1:	Layer 2:
Activation functions:	Hyperbolic tangent	Linear	Hyperbolic tangent
Input variables:	$Q(t\text{-}1)$, $\Delta P(t\text{-}1)$, $O_v(t)$	$Q(t\text{-}1)$, $\Delta P(t\text{-}1)$	$O_v(t)$
No. of hidden nodes:	10	1	10
Degree of freedom:	51	45	
Output variables:	$Q(t)$	$Q(t)$	

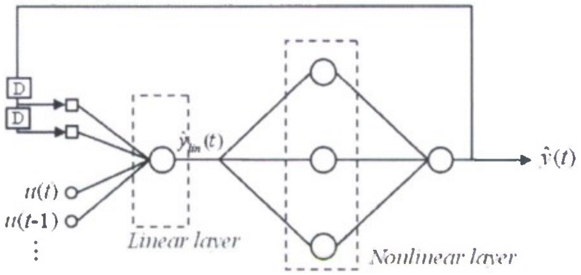


Figure 3: Block Structured NN

The comparison was done in a following way. A training data set and two test sets were generated from original computer model, with arbitrary changes on both *Ov* and ΔP over time. For each NN structure, five NN-based surrogate models were made using the same training set. For training of each model, 500 epochs and 1×10^{-6} MSE were set as the stopping conditions, and the training process stopped when any of the two conditions met. As performance is measured, MSE from one of the two test sets was measured to evaluate model approximation accuracy, and the training time and the number of epochs are used to assess training efficiency. All the NN implementation was made in Matlab®, and Levenberg-Marquardt method was chosen as the training algorithm. The results are shown in Table 3 and Table 4.

Table 3: Training Result of Plain NN

Trial No.	1	2	3	4	5	Average	Best
Training set MSE:	3.5609×10^{-6}	8.8035×10^{-6}	4.6208×10^{-6}	3.4656×10^{-6}	3.2052×10^{-6}	4.7312×10^{-6}	3.2052×10^{-6} (No. 5)
Test set MSE:	3.0385×10^{-4}	3.1160×10^{-4}	2.6629×10^{-4}	2.9953×10^{-4}	2.8758×10^{-4}	2.9377×10^{-4}	2.6629×10^{-4} (No. 3)
Training time (sec):	104.8	104.4	104.4	104.4	104.4	104.5	104.4
Epochs:	500	500	500	500	500	500	500
Training stopped by:	Max. epochs	Max. epochs	Max. epochs	Max. epochs	Max. epochs		

Table 4: Traing Result of Block-Structured NN

Trial No.	1	2	3	4	5	Average	Best
Training set MSE:	2.7326×10^{-6}	1.7462×10^{-6}	9.9629×10^{-7}	9.9897×10^{-7}	9.9347×10^{-7}	1.4935×10^{-6}	9.9347×10^{-7} (No. 5)
Test set MSE:	2.1417×10^{-4}	7.2219×10^{-5}	9.4172×10^{-4}	9.0177×10^{-5}	1.1590×10^{-4}	2.8704×10^{-4}	7.2219×10^{-5} (No. 2)
Training time (sec):	37.2	37.3	15.9	28.1	6.2	24.9	6.2
Epochs:	500	500	212	371	80	332.6	80
Training stopped by:	Max. epochs	Max. epochs	Error criterion	Error criterion	Error criterion		

Based on the results above, it indicates that the NN models with the block-structure seems to outperform the plain NN models in both training efficiency and model accuracy, except that the average values of the test set MSE almost tie though the best test set MSE of the block-structured NNs was about 3.7 times better than that of the plain NNs.

Thus, in order to further observe the performance in model accuracy, another test set was employed for simulation using the models from the two different structures. To have a better visualization of the simulation results, only the NN models with the best training set MSE and the best test set MSE were picked from each of the two groups to demonstrate the simulation tests. The plots of the simulation results are presented in Figure 4 and Figure 5.

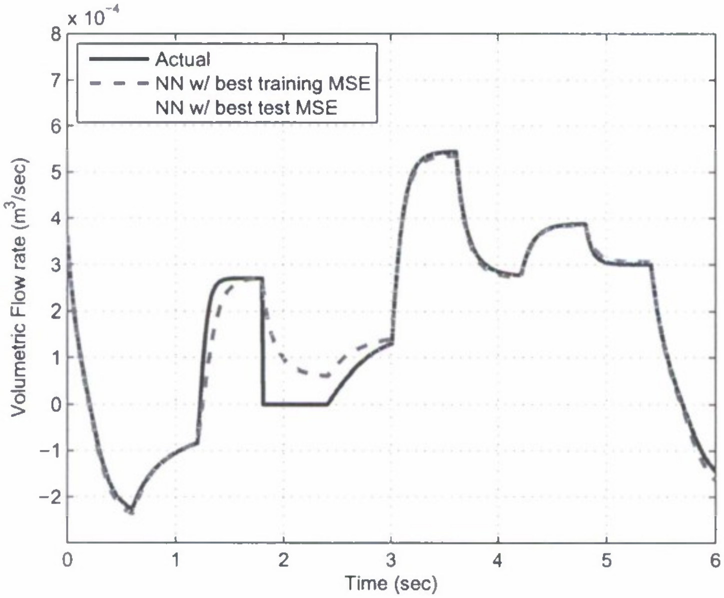


Figure 4: Simulation Result of Plain NN Models, Using another Test Set

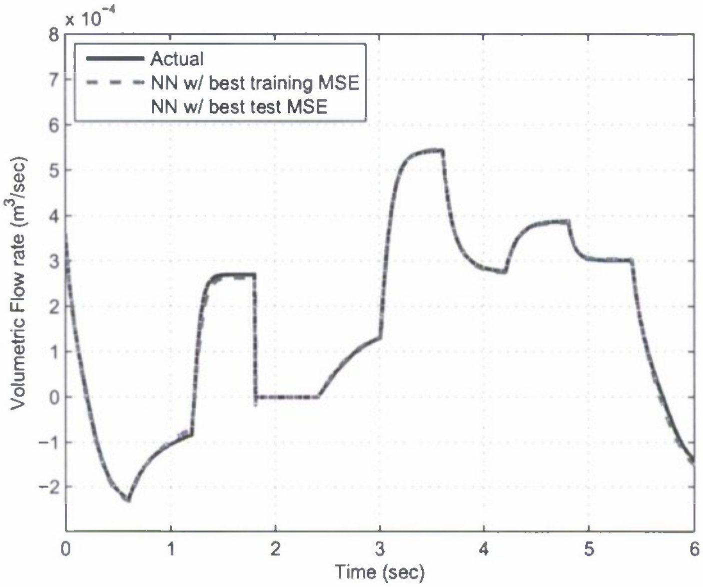


Figure 5: Simulation Result of Block-Structured NN Models, Using another Test Set

Unlike the previous results obtained through the simple RLC circuit model case, the plain NN models provided fairly good model stability and simulation accuracy, which may be because most fluid-system components have the inherent monotonic, fist order-like dynamics. However, as shown in Figure 4, the model also had strangely high errors in the period from about 1.2 seconds to 3 seconds of the simulation time, where the valve opening ratio, one of the inputs to the model, was relatively low.

On the other hand, the NN models with the block-structure delivered very good simulation accuracy over the entire simulation time, as can be seen in Figure 5. However there was one odd aspect in the result of the block-structured NN models too, which was the high overshoot of the model output at about 1.7 second of the simulation time where the valve was suddenly closed completely. With a few more manual tests, it was found that such a high pitch error could disappear or become negligible if the valve input was changed more gradually than a sudden step-type change to the complete closure. Expecting such an unrealistic step input is not applied in any actual simulation run, these models may be still valid enough to use. If still necessary, corrections can be made easily by many ways, one of which is just including a rule such as '$Q(t) = 0$ if $O_v(t) = 0$' that overwrites the result from the models.

Task 4: Human in the Loop Control

After an investigation of several software frameworks a suitable framework has been selected and a migration plan put in place. Given the following criteria the most appropriate framework is a product from Adobe called Flex. The product is a mixture of the Adobe Action script programming language and a markup language call MXML. Flex applications can be complied into byte code compatible with the popular Flash Player.

- Built in cross platform compatibility.
- More advaneed support for visual effects as an enabler of visual analyties.
- Additional support for a morc modular software design.
- Capable of fast, responsive, and intricate interfaces.

In addition to the selection of the software framework, a database engine has been sclected to facilitate the information management of thc IRIS systcm simulation and design. MySQL has shown in a simple proof of eoncept that it is more than capable of handling the load demands of the simulation cnvironment. It is simple to integrate with a variety of frameworks and platforms, and touts the strengths of the struetured qucry language. Currently it is onc of the most commonly used database cngines. It is possiblc that its suitability can change over time as Oracle has recently acquired MySQL. PostgreSQL has been seleeted as a fall back option. As a measurc to ensurc compatibility with both choices, fcature usage has been earefully selected as the HMI database libraries are being written.

Task 5: A Methodology for Improving the Mission Effectiveness in Complex Systems Design

One of the main objeetives of IRIS is to deliver a eoneeptual design methodology for more survivable and mission effeetive ships. There can be different underlying philosophies, based on which an improved design solution can be returned. Traditional design approaches would foeus on performanee, while modern approaehes will seek for morc robust solutions, either through enhancing safety or adding automation or intelligenee. Beyond the development of methods that allow for the diseovery of sueh solutions, studies ean also be proposed to investigate the tradcoff of cost vs. effeetiveness aeross solutions representing different underlying design philosophics.

Traditional design approaehes are based on optimizing naval system architccture for pcrformance, based on a very limited number of mission seenarios. However, such point solutions will only yield responses that maximize survivability mostly for scenarios similar to the ones that have been used for pcrformance optimization.

While the traditional design approach is eoneeptually fairly simple and straightforward, it does not really address any issucs regarding the uncertainty around naval system mission requirements, cnvironmental condition or even thc capability of the system to perform as designed under real operations. Moreover, it cannot guarantee that a feature that is an absolute best for a particular threat and operational situation, might not bc the best against the range of threats and operational situations it may eneounter. A robust solution will rcprcscnt a system that in theory would be better prepared to perform multiple mission aets and withstand a larger speetrum of unexpeeted events. At the same time, preseribed design performanee might not be optimal, in ordcr to compensate for the multi-mission eapability (e.g., preferred extra weight for redundant systems over maneuvcrability).

It is quite certain that robust design can offer system designs that are capable of surviving under various mission tasks and hazardous threat environments. However, the question at this point is how exactly the multi-mission capability and the enhanced survivability are enabled. Typical survivability enhancement features, such as component redundancy, separation and shielding are immediate techniques that can be properly applied to the design based on conceptual sizing. Real time simulations of systems operations can be also available for the sizing and decision making on selecting system architectures. This is still a form of robust design, yet through a more reactive approach to how hazards and environmental uncertainty affect system effectiveness.

A new philosophy has been recently emerging and seeks to address the aforementioned concerns. Resilience engineering is a novel and relatively recent form of philosophy on understanding threats, accident and damage propagation, as well as how a system should be designed to conform to changes that occur around it, for the purpose of withstanding adverse effects and maintaining its mission effectiveness. In other words, a resilient system can adjust its functioning prior to or following changes and disturbances so that it can go on working even after a major mishap or in the presence of continuous stress, mainly by being able to be proactive on safety.

Resilience engineering can offer insight and research directions that may lead to answers regarding the design of more safe and survivable complex systems. According to the systemic view of how accidents occur, one can infer that a resilient response by the system would include the ability to efficiently adjust to non-favorable influences rather than to resist them. Such ability could be embedded as collection of internal functionalities and be the basis for certain active features for susceptibility/vulnerability reduction and recoverability increase. Automation and networks of sensing grids and information distribution might be possible enablers for enhanced reconfigurability and would lead to the essential functionality of a resilient system.

The overall problem though, relies on investigating possible methods for improving system and mission effectiveness. According to the Defense Appropriation Act of 2004, effectiveness can be improved by including survivability in the design process as a key performance parameter. The current United States Navy standard is primarily determined by the Survivability Design Handbook for Surface Ships (OPNAV P-86-4-99). According to this procedure, survivability is improved by focusing on vulnerability and applying standard design principles such as subsystem redundancy or separation. Other common tools that are employed are the deactivation diagrams that are similar to fault tree diagrams in reliability studies.

The fundamental research question regarding this initiative would be how to improve the design the system, so that system effectiveness through survivability is maximized for a given set of scenarios, which will include system damage and/or restoration events. Moreover, it can extend to consider how the philosophy of resilience engineering can translate into a systems engineering method, involving various aspects, such as accident and damage modeling or system functionality and possible enablers, in order to fit into the bigger picture of more survivable systems in a highly uncertain mission environment.

Based on earlier work, it is assumed that there has been a clarification of how a robust solution would differ from the resilient solution. Most of current ship design methods are based on traditional design methods, yet robustness of solution is ensured through optimizing the architecture for multiple scenarios. However, nothing ensures that a robust solution is obtained with the use of a systemic accident model or, or that no significant excess of weight has been added due to the highly redundant subsystem components. But, more importantly, it is quite challenging to demonstrate how a robust architecture can be proactive through its embedded functionality, to be also recognized as a resilient system.

Furthermore, it is assumed that a first iteration of a method for resilient systems design will have been formulated. The main focus for this task should become the fine-tuning of this method and demonstrating its superiority if compared to the robust solution for the same mission scenario. At the same time, it is expected that there will be a cost-effectiveness tradeoff that could be investigated through the assessment of safety and survivability improvements against any performance degradation for both design approaches.

Based on the objectives stated earlier, a central hypothesis can be formulated and the proposed subtasks will aim towards supporting it. The hypothesis states the following:

A more resilient system demonstrates improved survivability than a robust system, mainly by incorporating engineering system reconfigurability, if subject to the same intelligent or natural events that affect system operations.

Improved safety and survivability come at some expense in overall system performance, acquisition and maintenance costs.

Before providing the outline of tasks that is combined should support this hypothesis, the following implications can be extracted:

- Robust systems can be survivable mainly through reduced vulnerability, yet without significant potential in active susceptibility reduction and recoverability enhancement.
- Resilience engineering suggests a collection of modern concepts that could potentially improve active survivability, mainly through the development of reconfigurable systems.
- While resilient systems are expected to be more survivable than robust systems under the same threat environment, it might be that such benefit will be at the cost of degraded system performance and higher acquisition and maintenance costs

Optimize two system architectures, using robust and resilient systems design respectively

With this subtask, two different approaches are adopted for delivering two alternative optimized solutions, starting from the same baseline. The common baseline is a version of a Yard Patrol craft (YP) that will be augmented for survivability improvement through suseeptibility, vulnerability and reeoverability reduetion. A general template of the method adopted is presented with Figure 2.

The robust design approaeh will mostly foeus on vulnerability reduetion, through the usage of more reaetive teehnologies and naval arehiteeting enhaneements. Some of them are involving redundant eomponents, strategie plaeement, sophistieated zonal design, lighter materials and enhaneed shielding. To a great extent, robust design is traditional naval arehiteeting, with improved systems engineering to satisfy more stringent safety requirements and deeision making for on seleeting the optimal solution, based on multi-mission operations simulation.

The resilient design approaeh is a robust solution to a great extent, yet it requires the system to be more proaetive for withstanding and reeovering from a threat and its resulting events. There ean be various enablers that will offer this eapability to the system. Reeonfigurability seems to be the most feasible alternative for making a system more resilient. This ean be aehieved with controllers that will support a series of automated funetions for sensing, analyzing and seleeting an appropriate plan for withstanding and neutralizing the effeets of the threat. Moreover, the implementation of a systemie aeeident and damage propagation model through real time modeling and simulation ean be instrumental in identifying additional modes of failure and damage that ean be taken into aeeount in the survivability assessment and improvement of the architecture.

The experimentation and design framework has been struetured to support design spaee exploration, systemie damage and aeeident modeling, physies based simulation for eapturing system behavior and ineludes “plaeeholders” for importing different reeonfiguration strategies through intelligent algorithms and seleeting the most suitable for a given arehiteeture.

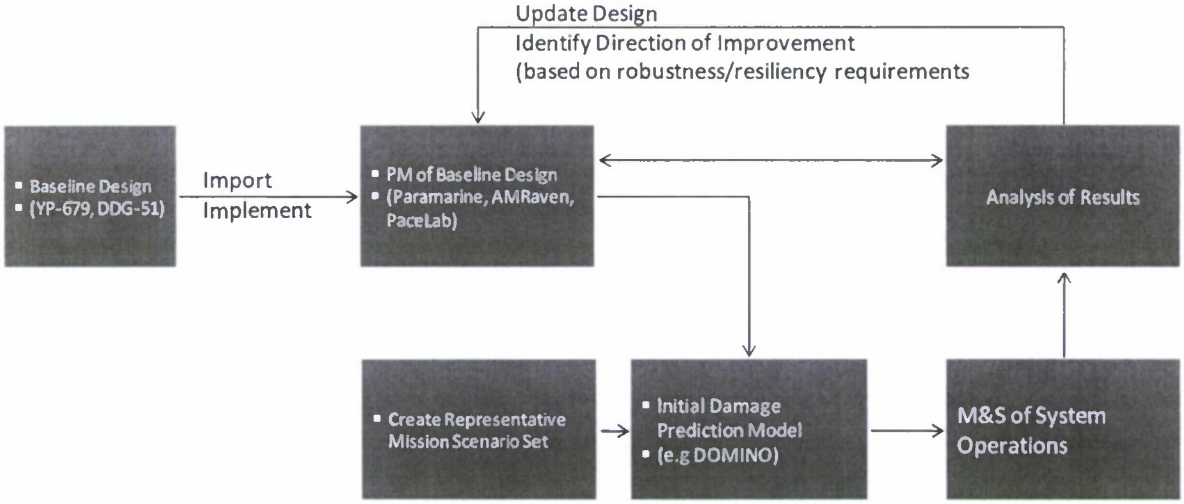
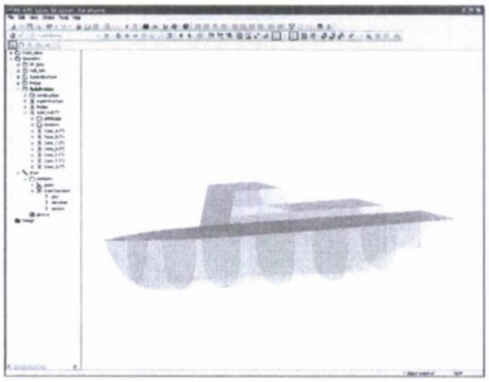
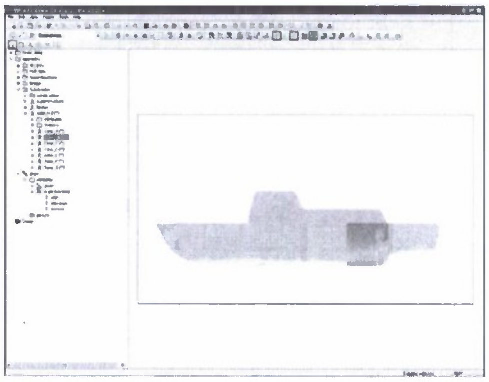

Update Design
Identify Direction of Improvement
(based on robustness/resiliency requirements
▪ Baseline Design
▪ (YP-679, DDG-51)
Import
Implement
▪ PM of Baseline Design
▪ (Paramarine, AMRaven, PaceLab)
Analysis of Results
▪ Create Representative Mission Scenario Set
▪ Initial Damage Prediction Model
▪ (e.g DOMINO)
M&S of System Operations

Figure 6: General template of robust and resilient design methods

As a baseline, a notional naval ship design is required to be the starting point for the implementation of the method. A synthesis and sizing tool is used for generating the geometry and the inner systems distribution. Paramarine is the software that has been used to create this baseline. It would require a certain amount of information for the creation of a ship baseline, such as ship geometry, engineering subsystems, acquisition and operations cost breakdown, mission profiles, threats and hazards and local environmental conditions.

The damage prediction module is responsible for analyzing and visualizing the damage propagation throughout the particular architecture. Based on the three different types of accident modeling, this module is more of a combination of a linear damage model and systemic. It is using DOMINO, a tool based on the theory of deactivation diagrams for initial damage prediction, based on the given single points of failure and subsystem connectivity. This module is linked to an M&S environment that simulates the operations of the ship's engineering plant, including the power generation and the cooling system with their corresponding controllers. In other words, the physics-based simulation represents the systemic model of failure prediction and is exchanging information with the deactivation diagrams at the end of every time step. Fire due to overheat and compartment flooding, are both expected to be part of damage modeling.





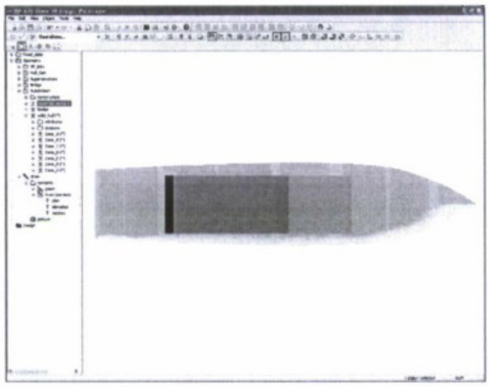


Figure 7: Screenshots of the ship geometry model for IRIS capability demonstration

A set of responses at different operational levels have been defined and require data that will be provided by the output of the simulation. Such metrics are the figures of merit for the particular design solution representing the corresponding architecture and will determine its performance based on survivability and mission effectiveness criteria. At the subsystem level, subsystem performance measures can be obtained (e.g. voltage outputs, coolant mass flow rates). Given a scenario per system configuration, system sensitivities and correlations of measures of performance (MoP) to scenario changes can be identified. Such measures are mostly conditional probabilities of achieving an outcome response, given events that occurred earlier, as defined by the scenario event tree analysis. By identifying the direction of improvement and exploring the design space, multiple iterations can be performed around the baseline to achieve a solution that satisfies the original design requirements.

Survivability mission effectiveness assessment is the next task that will enable the cost-effectiveness tradeoff for each solution. Despite the fact that some steps of a survivability assessment process have been already utilized for improving the solution at the design process, the objective of this task is to evaluate the complete solution. The template for the evaluation process is the Total Ship survivability Assessment Method (TSSA), an overview of which is provided at the Figure 8.

It should be expected however, that while the resilient solution demonstrates improvement in terms of safety and survivability, it might also incur increased development and maintenance costs. A similar case is expected for the robust solution, yet it should be investigated whether the extra costs for moving from a robust to a resilient system can justify the safety improvements and at what levels of mission performance degradation. For a mission with given outcomes, the integration of MoP to MoEs should look like Figure 9.

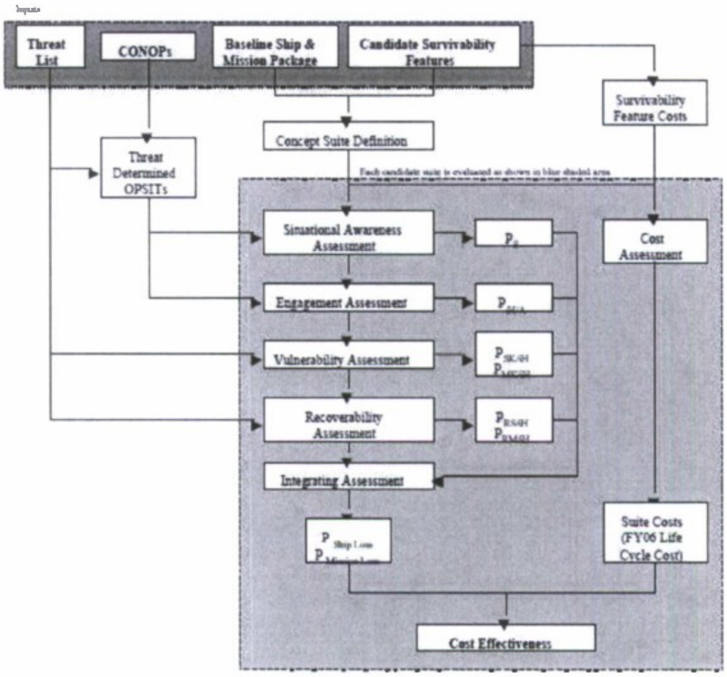

Inputs
Threat List
CONOPs
Baseline Ship & Mission Package
Candidate Survivability Features
Survivability Feature Costs
Concept Suite Definition
Threat Determined OPSITs
Each candidate suite is evaluated as shown in blue shaded area
Situational Awareness Assessment
P
Cost Assessment
Engagement Assessment
P
Vulnerability Assessment
P SK/H
P MK/H
Recoverability Assessment
P RS/H
P RM/H
Integrating Assessment
P Ship Loss
P Mission Loss
Suite Costs (FY06 Life Cycle Cost)
Cost Effectiveness

Figure 8: Total Ship Survivability Assessment Method (TSSA)

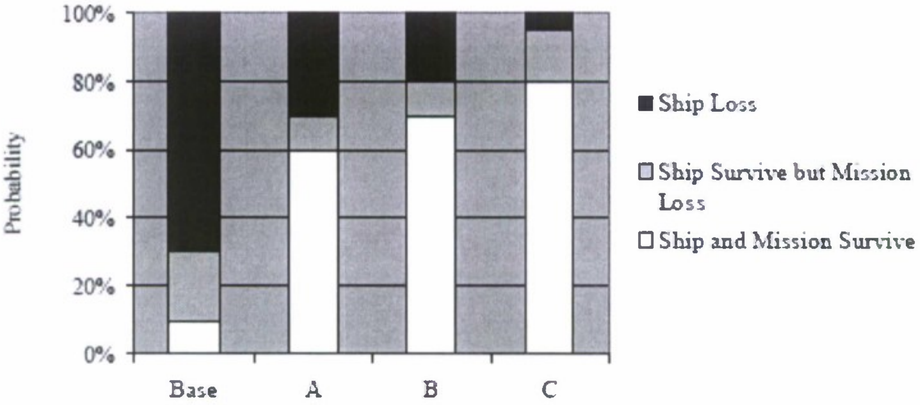


Figure 9: Probabilistic distribution of scenario outcomes as calculated by the TSSA method